2157
IFW

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/029,700	12/19/2001	Teddy Lindsey	028790.0012.UTL1	3885

23865 7590 08/16/2005

BROBECK, PHLEGER & HARRISON LLP
12390 EL CAMINO REAL
SAN DIEGO, CA 92130

EXAMINER	
NANO, SARGON N	
ART UNIT	**PAPER NUMBER**
2157	

DATE MAILED: 08/16/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	10/029,700	LINDSEY, TEDDY
	Examiner	Art Unit
	Sargon N Nano	2157

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 December 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☐ Claim(s) *1 - 66* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 - 66* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to application filed on Dec. 19 2001. Claims 1 – 66 are pending examination.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1 – 3, 5, 6, 8, 10, 14 – 18 - 26, 28, 29, 31 – 37, 39, 42 – 46, 49, 50, 53 - 56, 58 – 63 and 66 are rejected under 35 U.S.C. 102(e) as being anticipated by Britton et al U.S. Patent No.6,535,896 (referred to hereafter as Britton).

Britton teaches the invention explicitly as claimed. Britton teaches systems , methods and computer program products for utilizing XML – based tools to tailor HTML – based web page content for display within various client devices (see abstract).

As to claim 1, Britton teaches a method of transcoding information in a first markup language into a second markup language, the method comprising the steps of:

responding to a request to view a Web page by retrieving information from said Web page, wherein said information is in a first markup language (see col. 3 lines 28 – 39, Britton discloses requesting an HTML- based web page) ;

normalizing said information(see col. 3 lines 28 – 43 Britton discloses transferring the HTML – based web page format to XML format);

determining a second markup language that can be used by a browser using device detection, wherein said browser is used by a computer that is to view said information (see col. 3 lines 40 – 48,Britton discloses the modified portions of XML document are converted back to HTML format) ; and

transcoding said information into said second markup language(see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 2, Britton teaches the method of claim 1, further comprising the step of sending said information in said second markup language to said computer (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 3, Britton teaches the method of claim 2, wherein said computer is a wireless mobile device (see col. 3 lines 28 - 38. Britton discloses sending of a web page to pervasive computing devices).

As to claim 5, Britton teaches the method of claim 2 wherein the step of sending said information in said second markup language to said computer comprises sending said information to said computer using automatic page division (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 6, Britton teaches the method of claim 1, wherein said step of transcoding comprises the steps of:

selecting a renderer that is associated with said second markup language from a plurality of renderers associated with markūp languages (see col.3 lines 43 – 48, Britton discloses converting back to HTML format);

sending said information through said renderer (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display); and

transcoding said information into said second markup language using said renderer(see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format).

As to claim 8, Britton teaches the method of claim 1, wherein said step of normalizing comprises the step of transcoding said information in said first markup language into an intermediate markup language (see col. 5 lines 1 – 11 Britton discloses that XSL can be used to transform an XML document into one HTML view).

As to claim 10, Britton teaches the method of claim 1, wherein said second markup language comprises the EXtensible Markup Language (XML) (see col.6 lines 10 – 14 Britton discloses HTML web page content format converted to XML format).

As to claim 14,Britton teaches the method of claim 1, wherein said second markup language comprises the HyperText Markup Language (HTML) (see col. 3 lines 40 49, Britton teaches HTML language).

As to claim 15, Britton teaches the method of claim 1, wherein said steps of responding, normalizing, determining, and transcoding occur automatically (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 16, Britton teaches the method of claim 1, wherein said first markup language comprises the HyperText Markup Language (HTML) (see col. 3 lines 40 49, Britton teaches HTML language).

As to claim 17, Britton teaches the method of claim 1, further comprising the step of sending said information in said second markup language to said computer over a system of networked computers (see fig. 2 Britton discloses a pervasive computing device in a network).

As to claim 18, Britton teaches the method of claim 1, wherein a first object embodies said information in said first markup language and said step of transcoding further comprises automatic object conversion of said first object to a second object embodying said information in said second markup language (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format).

As to claim 19, Britton teaches the method of claim 1, further comprising providing an error logging system (see col. 4 lines 54 – 60 Britton discloses reporting errors).

As to claim 20, Britton teaches the method of claim 1, wherein said second markup language is a markup language other than the HyperText Markup Language (HTML) (see col. 5 lines 1 – 13 Britton discloses Extensible Stylesheet Language).

As to claim 21, Britton teaches the method of claim 1, wherein said device detection comprises referring to an HTTP user agent header field (see col. 6 line 66 – col. 35, Britton discloses the hypertext transfer protocol).

As to claim 22, Britton teaches the method of claim 1, wherein said device detection comprises detecting said browser and said computer using unique signature detection (See col. 3 lines 7 – 14, Britton discloses a modified web browser).

As to claim 23, Britton teaches the method of claim 1, further comprising dividing said information in said second language into at least two pages using automatic page division (see col. 3 lines 50 – 62 Britton discloses web pages having a mixture of formats to be converted to a single format).

As to claim 24, Britton teaches a method of transcoding information in a first markup language into a second markup language, the method comprising the steps of:

responding to a request to view a Web page via a computer (see col. 3 lines 28 – 39, Britton discloses requesting an HTML- based web page);

retrieving information from said Web page, wherein said information is in a first markup language, normalizing said information (see col. 3 lines 28 – 43 Britton discloses transferring the HTML – based web page format to XML format); and

transcoding said information into a second markup language, wherein said computer is adapted for utilizing said second markup language(see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 25, Britton teaches the method of claim 24, wherein said step of normalizing comprises the step of transcoding said information in said first markup language into an intermediate markup language (see col. 5 lines 1 – 11 Britton discloses that XSL can be used to transform an XML document into one HTML view).

As to claim 26, Britton teaches the method of claim 24, wherein said computer is a wireless mobile device (see col. 3 lines 28 - 38. Britton discloses sending of a web page to pervasive computing devices).

As to claim 28, Britton teaches the method of claim 24, further comprising dividing said information in said second language into pages using automatic page division (see col. 3 lines 50 – 62 Britton discloses web pages having a mixture of formats to be converted to a single format).

As to claim 29, Britton teaches the method of claim 24, wherein said step of transcoding comprises the steps of:

determining said second markup language, wherein said computer is adapted for utilizing said second markup language(see col. 3 lines 40 – 48,Britton discloses the modified portions of XML document are converted back to HTML format);

selecting a renderer that is associated with said second markup language from a plurality of renderers associated with markup languages(see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format);

sending said information through said renderer (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display) ; and

transcoding said information into said second markup language using said renderer (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format).

As to claim 31, Britton teaches the method of claim 24, wherein said steps of responding, retrieving, normalizing, and transcoding occur automatically (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 32, Britton teaches the method of claim 24, wherein a first object embodies said information in said first markup language and said step of transcoding further comprises automatic object conversion of said first object to a second object embodying said information in said second markup language (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format).

As to claim 33, Britton teaches the method of claim 24, further comprising providing an error log that reports errors that occur during at least one of said steps of responding, retrieving, normalizing, and transcoding (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 34, Britton teaches the method of claim 24, wherein said second markup language is a markup language other than the HyperText Markup Language (HTML) (see col. 5 lines 1 – 13 Britton discloses Extensible Stylesheet Language).

As to claim 35, Britton teaches the method of claim 24, further comprising the steps of: detecting a browser of said computer; and determining said second markup

language that is used by said browser based on said step of detecting (see col. 6 line 66 – col. 35, Britton discloses the hypertext transfer protocol).

As to claim 36, Britton teaches a method of transcoding information in a first markup language into a second markup language, the method comprising the steps of:

responding to a request to view a Web page language (see col. 3 lines 28 – 39, Britton discloses requesting an HTML- based web page);

retrieving information from said Web page, wherein said information is in a first markup language(see col. 3 lines 28 – 43 Britton discloses transferring the HTML – based web page format to XML format);

device detection to determine said second markup language that is used by said browser(See col. 3 lines 7 – 14, Britton discloses a modified web browser) ; and

transcoding said information into a second markup language, wherein said computer is adapted for utilizing said second markup language(see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 37, Britton teaches the method of claim 36, wherein said computer is a wireless mobile device (see col. 3 lines 28 - 38. Britton discloses sending of a web page to pervasive computing devices).

As to claim 39, Britton teaches the method of claim 36, wherein said step of transcoding comprises the steps of:

selecting a renderer that is associated with said second markup language from a plurality of renderers associated with markup languages(see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format);

sending said information through said renderer (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display) ; and

transcoding said information into said second markup language using said renderer (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format).

As to claim 41, Britton teaches the method of claim 36, wherein said steps of responding, retrieving, device detection and transcoding occur automatically (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 42, Britton teaches the method of claim 36, further comprising dividing said information in said second language into pages using automatic page division (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 43, Britton teaches the method of claim 36, wherein a first object embodies said information in said first markup language and said step of transcoding further comprises automatic object conversion of said first object to a second object embodying said information in said second markup language (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format).

As to claim 44, Britton teaches the method of claim 36, further comprising transcoding said information in said first markup language into an intermediate markup language prior to transcoding said information into second markup language (see col. 5 lines 1 – 11 Britton discloses that XSL can be used to transform an XML document into one HTML view).

As to claim 45, Britton teaches a system for viewing a Web page by a computer that utilizes a markup language, the system comprising:

a computer, wherein said computer requests to view a Web page(see col. 3 lines 28 – 39, Britton discloses requesting an HTML- based web page);

information from said Web page, wherein said information is in a first markup language; a device detector, wherein said device detector determines a second markup language that said computer utilizes(see col. 3 lines 40 – 48,Britton discloses the modified portions of XML document are converted back to HTML format); and

a renderer, wherein said renderer transcodes said information into said second markup language, wherein said information is sent to said computer(see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 46, Britton teaches the system of claim 45, further comprising:

a normalizer, wherein said normalizer transcodes said information in said first markup language into an intermediate markup language(see col. 3 lines 28 – 43 Britton discloses transferring the HTML – based web page format to XML format).

As to claim 49, Britton teaches the system of claim 45, wherein said computer utilizes a markup language other than the HyperText Markup Language (HTML) (see col. 5 lines 1 – 13 Britton discloses Extensible Stylesheet Language).

As to claim 50, Britton teaches the system of claim 45, wherein said computer is a wireless mobile device (see col. 3 lines 28 - 38. Britton discloses sending of a web page to pervasive computing devices).

As to claim 53, Britton teaches the system of claim 45, wherein said information in said second markup language is sent to said computer over a system of networked computers (see fig. 2 Britton discloses a pervasive computing device in a network).

As to claim 54, Britton teaches the system of claim 45, wherein a first object embodies said information in said first markup language and said renderer uses automatic object conversion to convert said first object to a second object embodying said information in said second markup language (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format).

As to claim 55, the system of claim 45, further comprising an error logging system (see col. 4 lines 54 – 60 Britton discloses reporting errors).

As to claim 56, Britton teaches the system of claim 45, wherein said second markup language is a markup language other than the HyperText Markup Language (HTML) (see col. 5 lines 1 – 13 Britton discloses Extensible Stylesheet Language).

As to claim 57, Britton teaches the system of claim 45, wherein said device detector uses unique signature detection (See col. 3 lines 7 – 14, Britton discloses a modified web browser).

As to claim 58, Britton teaches a system for viewing a Web page by a computer that utilizes a markup language other than the HyperText Markup Language (HTML), the system comprising:

a computer, wherein said computer requests to view a Web page (see col. 3 lines 28 – 39, Britton discloses requesting an HTML- based web page) ;

information from said Web page, wherein said information is in a first markup language(see col. 3 lines 28 – 39, Britton discloses requesting an HTML- based web page);

a normalizer, wherein said normalizer normalizes said information in said first markup language into an intermediate markup language(see col. 3 lines 28 – 43 Britton discloses transferring the HTML – based web page format to XML format); and

a renderer, wherein said renderer transcodes said information in said intermediate markup language into a second markup language, wherein said second markup language is a markup language that said computer utilizes and said second markup language is a markup language other than HTML (see col. 5 lines 1 – 11 Britton discloses that XSL can be used to transform an XML document into one HTML view).

As to claim 59, Britton teaches the system of claim 58, further comprising:

a device detector, wherein said device detector determines said second markup language based on a browser of said computer (see col. 6 line 66 – col. 35, Britton discloses the hypertext transfer protocol).

As to claim 60, Britton teaches the system of claim 58, wherein said computer is a wireless mobile device (see col. 3 lines 28 - 38. Britton discloses sending of a web page to pervasive computing devices).

As to claim 61, Britton teaches the system of claim 58, wherein a first object embodies said information in said first markup language and said renderer uses automatic object conversion to convert said first object to a second object embodying said information in said second markup language (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML format).

As to claim 62, Britton teaches computer executable process steps operative to control a computer, stored on a computer readable medium, comprising:

a plurality of steps to receive data required for subsequent calculations(see col. 3 lines 28 – 43 Britton discloses transferring the HTML – based web page format to XML format); and

a plurality of steps to automatically transcode information in a first markup language into a second markup language, wherein said second markup language is automatically determined (see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

As to claim 63, Britton teaches the steps of claim 62, further comprising a step to automatically normalize said information in said first markup language prior to transcoding said information into said second markup language (see col. 3 lines 28 – 43 Britton discloses transferring the HTML – based web page format to XML format).

conversion of the document to HTML format).

As to claim 66, Britton teaches the method of claim 64, further comprising dividing said information in said second language into pages using automatic page division (see col. 3 lines 50 – 62 Britton discloses web pages having a mixture of formats to be converted to a single format).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 4, 7, 27, 30, 38, 40, 47, 48, 51, 52, 64 and 65 are rejected under 35 U.S.C. 103(a) as being unpatentable over Britton in view of Vega – Garcia et al. U.S.Patent No. 6,839,734 (referred to hereafter as Vega-Garcia).

As to claims 4, 7, 27, 30, 38, 40, 47, 48, 51 and 52, Britton teaches systems , methods and computer program products provided utilizing XML-based tools to tailor

HTML-based Web page content for display within various client devices. Moreover Britton teaches a method of transcoding information in a first markup language into a second markup language, the method comprising the steps of:

responding to a request to view a Web page by retrieving information from said Web page, wherein said information is in a first markup language (see col. 3 lines 28 – 39, Britton discloses requesting an HTML- based web page) ;

normalizing said information(see col. 3 lines 28 – 43 Britton discloses transferring the HTML – based web page format to XML format);

determining a second markup language that can be used by a browser using device detection, wherein said browser is used by a computer that is to view said information (see col. 3 lines 40 – 48,Britton discloses the modified portions of XML document are converted back to HTML format) ; and

transcoding said information into said second markup language(see col. 3 lines 40 – 48, Britton discloses the conversion of the document to HTML and sent the web page with modified content to a client device for display).

Britton does not teach streaming information in real time. However, Vega-Garcia teaches multimedia communications software with network streaming and multi-format conference (see col. 4 lines 20 – 27). It would have been obvious to one of the ordinary skill in the art to include streaming information in real time as in Britton's because doing so would enable a user to avoid delay entailed in downloading an entire file and then playing it .

As to claims 64 and 65, Britton teaches the invention as mentioned above. Britton does not teach streaming information in real time. However, Vega-Garcia teaches multimedia communications software with network streaming and multi-format conference (see col. 4 lines 20 – 27). It would have been obvious to one of the ordinary skill in the art to include streaming information in real time as in Britton's because doing so would enable a user to avoid delay entailed in downloading an entire file and then playing it .

4. Claims 9 and 11-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Britton.

Britton teaches the intermediate markup language is XML and the second format is Html. Britton does not explicitly teach the intermediate markup language is XHTML, and the second markup language is WML, cHTML or HDML. Official notice is taken that it would have been obvious for one of the ordinary skill in the art at the time of the invention was made to modify Britton by using the markup languages because doing so allow the system to use different formats of markup languages that are extremely simple" dialect of language suitable for use on the World-Wide-Web and therefore does not require use of specialized software on the client device.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- System and Method for Cooperative Client/Server Customization of Web pages by Fields et al. U.S. Patent No. 6,412,008.

- System In which a Proxy- Server Translates Information Received From the Internet Into A Form/Format readily Usable By Low Power Portable computers by Kiknis U.S. Patent No. 5,727,159.
- System For Dynamically Transcoding Data Transmitted Between Computers by Tso et al U.S.Patent No. 6,421,733
- An Active Transcoding Proxy to Support Mobile Web Access by Harini Bharadvaj (University of Missouri- Columbia).

Transcoding Internet Content For Heterogeneous Client Devices. By John R. Smith, Rakesh Mohan and Chung- Sheng Li, pages III-599 – III- 602.Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sargon N Nano whose telephone number is (571) 272-4007. The examiner can normally be reached on 8 hour.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571) 272-4001. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sargon Nano

March 10, 2005

ARIO ETIENNE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Notice of References Cited	Application/Control No. 10/029,700	Applicant(s)/Patent Under Reexamination LINDSEY, TEDDY	
	Examiner Sargon N Nano	Art Unit 2157	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,535,896	03-2003	Britton et al.	715/523
	B	US-6,839,734	01-2005	Vega-Garcia et al.	709/204
	C	US-5,727,159	03-1998	Kikinis, Dan	709/246
	D	US-6,421,733	07-2002	Tso et al.	709/246
	E	US-6,412,008	06-2002	Fields et al.	709/228
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	An Active Transcoding Proxy to support Mobile web Access, By Harini Bharadvaj, Joshi sansanne Auephanwiriyakul of University of Missouri
	V	Transcoding Internet Co ntent For Heterogenous Client Devices by John R.Smith, rakesh Mohan and chung- sheng Li, pp III - 599 - III - 602
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

An Active Transcoding Proxy to Support Mobile Web Access*

Harini Bharadvaj, Anupam Joshi and Sansanee Auephanwiriyakul
Department of Computer Engineering & Computer Science,
University of Missouri-Columbia,
Columbia, MO 65211 USA
E-mails: {harini,sansanee}@meru.cecs.missouri.edu, joshi@cecs.missouri.edu

Abstract

In this paper, we present a proxy based system (MOWSER) to support web browsing from mobile clients over wireless networks. Mowser is a proxy agent between the mobile host and the web server, which performs active transcoding of data on both upstream and downstream traffic to present web information to the mobile user according to the QoS parameters set by the user. Active transcoding is defined as modifying the HTTP stream in situ, and it is entirely transparent to the user. Further, our system does not pose any additional requirements on the mobile user. This is an improvement over other proxy based systems, which only transcode images on the downstream and are mostly not configurable. While developed for mobile users, such a system can actually be useful in any low bandwidth scenario.

1. Introduction

With an ever increasing amount of information "out there" on the World Wide Web, and mobility becoming the need of the hour, users want to have information at their fingertips wherever they may be. Recent developments in mobile computing and web technologies have resulted in an increase in the number of "road warriors", people who use mobile computers to access their business information repositories, often through the web. These users need speedy access to that data. One cannot expect them to wait for hours to download a web page containing a lot of multimedia data over a wireless network, most of which cannot be handled by the small and "easy-to-carry" laptop/palmtop computer anyway. Most mobile computers have very limited cpu, memory & disk resources. They communicate over wireless links which are characterized by lower bandwidths, higher error rates, and more frequent disconnections. Moreover, in a mobile environment, changes to the network bandwidth and resources are very common. These are only some of the challenges of mobile computing, more of which are discussed in [4]. In the context of web browsing, web servers do not consider the network connection between them and the client. They also have no concern for the hardware capabilities of the client. They just return the document asked for assuming that the client is capable of properly receiving and displaying the data. People are commonly creating web pages rich in multimedia data, pages with several images and videos have become very common. Presenting all this multimedia-rich data over wireless networks to the mobile user is a major challenge. The mobile computer either does not have the appropriate hardware or adequate bandwidth (or both) to handle the content, and the web servers also cannot be modified to suit the mobile user. Therefore, the only way of bridging the gap between the highly resource-rich web servey at one end and the highly resource-poor mobile client at the other, is by introducing a system in between which modifies the contents on the web and present it in the most appropriate form to the mobile user. Hence, the need for middleware, which adapts to the mobile environment, gets the best of what is available on the web and does not pose any additional requirements on the mobile client.

We have devised such a middleware based solution which allows the user of a mobile computer to control the way in which the data from the web is retrieved, dynamically transform the data in a way that is transparent to the user, without requiring the mobile computer to do any additional work. We achieve this by introducing a proxy agent between the mobile client and the web server. Our proxy, Mowser, lets the user specify and control the viewing preferences and hardware capabilities of the client, transforms the data to and from the web server, without requiring any change on the client. We talk about related work in section 2 and describe the software architecture of our system in section 3. In section 4, we describe the various methods of transforming data that we have used. The results of our experiments are presented in section 5 and we discuss it in section 6.

2. Related Work

The Client-Proxy-Server model has begun to feature in many mobile applications to overcome the challenges faced in the mobile computing scenario. However, only some of them actually transform the stream between the client and the server.

In the GloMop model described in [5],[11] the proxy performs "distillation" of the document received from the server before sending it to the client. Distillation is defined here as highly lossy, real-time, datatype-specific compression that preserves most of the semantic content of a document. The approach of transcoding image files is similar to our approach first outlined in [9], but video files are handled

* This work was supported in part by an IBM Faculty Development Award to A. Joshi

differently. They perform real-time transcoding of motion JPEG to sub-sampled H.261, while our approach, described later in this paper, represents a video stream using representative frames. GloMop also allows refinement of selected portions of the document.

The Mowgli model [14] consists of two mediators, the Mowgli Agent and the Mowgli Proxy located on the mobile host and the mobile-connection host respectively. They use the Mowgli HTTP protocol to communicate with each other, which reduces the number of round-trips between client and server. A specialized transport service, the Mowgli Data Channel Service is used for reliable communication between the mobile-connection host and the mobile host. Mowgli WWW reduces the data transfer over the wireless link in three ways: data compression, caching, and intelligent filtering. It only performs GIF to JPEG conversion, and large embedded images are not transferred at all to the mobile node.

Zenel proposes an architecture in [19] in which the proxy is made up of three components: High Level Proxy, Low Level Proxy and Event Manager. The High Level Proxy allows filters for application layer protocols to be downloaded dynamically from mobile host applications. The Low Level Proxy is used to create and install filters for the transport and network layers. A control interface is provided for the filters running within these proxies by the Event Manager.

HTTP transducers called OreOs (defined in [3]) are specialized processing modules having four classes of functionality: filtering HTTP requests and responses, characterizing sets of messages, transforming message contents and additional processing indicated by the messages. The "stream model" approach of signal processing is used to perform complicated processing.

In the work of Satliyanarayanan et. al, a module called Cellophane [17] on the client transforms HTTP requests from Netscape into file operations on Odyssey [16] web objects and makes use of Odyssey API to select fidelity levels for images which are forwarded to a distillation server. The distillation server distills the images to the requested fidelity before passing them back to the client. However, this approach is specific to the Odyssey file system and requires a modified version of the Net BSD kernel. This also requires the addition of a module on the client.

Intel's Quick Web Technology [7] which sits on ISP servers compresses images by selectively dropping bits or pixels out of an image using lossy compression techniques, thereby speeding up the download of graphically- rich web pages. It also caches data to overcome the problem of bandwidth bulge. This can be used only when the access to the Internet is through ISP, and the user requires a Java-enabled browser to have control over the compression.

IBM's Web Express [6] consists of two components: ARTour (Advanced Radio Communications on Tour) Gateway and ARTour Client. The Gateway provides secure, compressed data across the selected network with authentication. It can automatically retrieve Web requests in the background while mobile users are performing other tasks.

In our system, the proxy performs active transcoding of HTTP requests from the client while sending it to the server, according to the preferences set by the mobile host, so that the document in the most suitable format is retrieved. It also processes the received HTTP data before sending it to the mobile host if necessary.

3. Software Architecture

Mowser is a proxy HTTP server agent (written in Perl) [18] which allows a mobile user to specify his or her viewing preferences, based on the network connection and available resources, and performs active transcoding of HTTP streams accordingly. The software architecture that we propose introduces a proxy server on each Mobile Support Station (MSS) to the basic MSS-MH (Mobile Host) model which accepts and stores the preferences for each of its mobile hosts, acts as the server to the mobile host, and as a client to the WWW server. No modifications to the web client on the MH is required. So any WWW browser that can handle forms and has the provision of a proxy can be used. No additional software is required on the mobile host. Also, setting and updating preferences is done by just filling up a CGI form on a URL at the web site maintained by the proxy server.

In the initial versions of Mowser [9],[10], the proxy dealt with getting the viewing preferences for a MH from the user and storing it according to its IP address. The current version also stores the accept headers that will be most suitable for the MH based on the preferences set by the user. The viewing preferences stored for each MH include a starting point, color capability, video resolution, sound capability, maximum allowed size for text, image, video, audio files and files of unknown type, and the size reduction technique for image files. Not all of these variables are presently used by the proxy. The preferences can be updated by the user of the MH whenever there is a change in the network connection or available resources. We are using an Apache HTTP server to store the preferences, and CGI scripts written in Tcl and Perl are used to update and save the preferences.

Once the MH sets the Proxy server as its proxy, all communication between the MH and the WWW servers is directed through this proxy. When the proxy receives a request from the MH, it looks up the preferences stored with the IP address of the MH and processes the request accordingly. Default preferences are used if no preferences had been specified. The proxy processes requests to set preferences and the GET requests. All other requests are forwarded to the target WWW server. In the next section, we detail how the proxy performs active transcoding of HTTP streams.

4. Active Transcoding

Modifying the HTTP stream and changing its content in situ is called active transcoding. This is done dynamically without any user intervention. For example, if an image file does not meet the size or color specifications, it is reduced before being sent to the MH (described in section 4.2). Similarly, sound files will not be sent to a MH with no sound capabilities, and so on.

Traditionally, transcoding is a unidirectional process [9], [5],[6],[7]. In other words, the request from the client is passed as is to the target server, while the return stream's multimedia content is altered. In our work, we alter the request as well, so as to take advantage of some net-friendly features of HTTP/1.1.

After setting preferences and making our MOWSER as its proxy, the user can browse the web as s/he would with any web client. On receiving a request from the MH, the proxy fetches the preferences set by the MH and serves the MH with files in the most suitable format. Default preferences are used if no values had been set by the MH. We

process HTTP GET requests received from the MH before sending it to the WWW server, and modify image and video files received from the WWW server before transmitting it to the MH if necessary. To support MHs with very limited resources and hardware capabilities like PDAs, we even parse the HTML stream to remove the active content and any tags that the MH cannot handle. The user may even choose to block any HTML file greater than a given size. With transcoding being done at two steps independently as shown in Figure 1, we are making sure that we match the preferences of the MH, while using the wired bandwidth in the most efficient manner.

4.1. Transcoding of HTTP Requests

HTTP/1.1 introduces the concept of content negotiation. The basic idea is that a WWW server may have several different representations of a resource. For example, it may store a document as postscript or word or HTML, etc. The server can automatically choose the file to send if the client sends the preferred representations as part of each request. Most servers (e.g., apache), even though not fully 1.1 compliant, already support content negotiation and will store files in several formats and in several variations of a format. We use this idea to get the file in the most appropriate format for the MH. For example, an image file may be made available in varying resolutions by the content provider on the server. We request the server to send the image file which has the resolution appropriate to the present QoS and client parameters, by including the preference in the request. This requires that the variants of a file have different mime types. For example, in our experiments we have used image/x-sgif to denote an image file with very low resolution, image/x-mgif to denote one with medium resolution and image/x-lgif to denote one with large resolution. We have also introduced video/x-rmpg to denote representative frames of video files (discussed in section 4.3).

Any HTTP GET request received from the MH is munged to an HTTP 1.1 request and the complete URI is included in the request line. The Accept headers stored for the MH are then appended to the outgoing stream to request for the file in the format most suited for the MH. A Host header is added to complete the HTTP 1.1 request. The server performs content negotiation and sends the file which closely meets the format specified in the request. Thus, the process is transparent to the user, and works even if the request comes from an HTTP/1.0 compliant browser, like most present commercial systems.

For example, for a MH host on a low bandwidth line, the proxy may append the following Accept headers to the request after making it a HTTP 1.1 request:

Accept: image/x-sgif, video/x-rmpg

For a MH like the PalmPilot, which can handle only text and images, the proxy greatly reduces the data transfer by selectively GETing the files. That is, when the proxy receives a GET request from a PDA, it sends a HEAD request to the WWW server to get information about the content type of the file, and then GETs the file only if the PDA can handle it. For example, the proxy does not request for audio, video and application files for a PDA. Since a page has the URLs of additional files to be fetched embedded in it, we could prevent the client on the MH from generating the additional GET requests and design the proxy to decide whether to GET the file or not by just looking at the extension of the file name in the embedded URLs. However, the content-type of the file is a better indicator of the format of the file than the extension in the filename, though getting this information requires an additional HEAD request to be sent.

4.2. Transcoding Image Files

When the proxy finds an image tag in the HTTP stream received from the server, it reads the URI of the image file to be fetched and first sends a HEAD request to the server. It checks the content-type and content-length information received from the server. If the content-length is small enough to be handled on the MH, the image file is sent to the MH unmodified. But if the image is larger than what can be handled by the MH, it is reduced in size or color as requested by the MH. The image files are scaled down in size, or the number of colors is reduced, or both without sacrificing semantics. On an image map for instance, size is not changed, only colors are, to preserve the semantics. The content-type information is used to decide the transformations that the image file has to go through. We convert all images to be reduced to portable pixmap format for processing and then convert them back to gif format for displaying. Then the original URL in the image tag is replaced with the URL of the modified image stored locally by the proxy and sent to the MH. This makes the MH GET the modified image file from the proxy. To display images on PDAs, the proxy might have to reduce images to 2-bit gray scale and thumb size.

4.3. Transcoding Video Data

Unlike image data where transcoding steps are obvious, video data represents a great challenge. Simple subsampling, as proposed in [5], is still not adequate as some clients may not have enough computational resources to do software decoding of MPEG or H.261. We use the structure inherent in video streams to do the transcoding. The structure of video is a hierarchy of the movie or episode. This hierarchy is segments, scenes, and shots. Each segment consists of sequence of scenes, each scene consists of several shots, and each shot is composed of several frames which have similar visual properties. Thus one of these frames can be selected as a Representative frame (Rframe) for the shot.

We present the video to the user by the representative frames which are picked from each shot. Using techniques, we have developed [1],[8] to support content based access to networked video databases. We have used several fuzzy clustering algorithms, such as fuzzy c-mean, hard c-mean, fuzzy c-median, hard c-median and possibilistic c-mean [2],[13],[12]. We use luminance and chrominance features, and 1-norm and 2-norm distance measures [1], [8] in order to group the frames which have similar properties together. Each group is classified as one shot. We pick the frame that is closest to center of each group to be Rframe. We do not explicitly use any scene change detection algorithms. The fuzzy techniques are used since frames can belong to the clusters to different degrees (membership values). Traditional scene change algorithms, which insist on a frame belonging to only one group, break down when confronted with gradual scene changes typically found in videos. While Rframes can be computed dynamically by the proxy, we feel that from a computational perspective, this should be done at the server side. In fact, it can be argued [1],[8] that these will typically be available at the server side already to support querying and browsing of the video database.

4.4. Transcoding Active Content

For mobile hosts with limited memory and computational resources, the user can decide not to receive any Java applets, JavaScripts, VBScripts, etc. When such a preference is set, the document to be transmitted is parsed and all the active content is eliminated before sending it to the MH. This is specifically suitable for the PDAs which have very small disk space and low speed CPUs. Also, given restrictions on the memory footprints of the applications that can reside on such machines, it is not clear that browsers will be able to support the virtual machines needed for active content languages such as Java.

Often though, Java (and JavaScript) are used to provide functionality that can be duplicated using CGI callbacks or server parsed HTML. An interesting option that we wish to pursue here is to see if we could have the forms capable client request for a CGI version instead of the JavaScript from the server. In other words, replace active content of a page with equivalent dynamic content. This feature will be supported via content negotiation. Like all content negotiation, it assumes that the server provides alternative versions (CGI based vs Java based) of a particular URL.

4.5. Transcoding HTML

Claim 8

We can reduce the computation on the MH by parsing HTML tags on the proxy itself, rather that on the MH. We can eliminate all the tags that the MH does not support, and references to any file that the MH is not capable of handling. For example, we can eliminate the italics tag, cascade style sheets, etc. for a PDA such as the PalmPilot. For such severely resource constrained MHs, the set of tags that it can handle may be so small, that it is advantageous to strip of all unwanted tags at the proxy, and encode the remaining tags using a few bits.

By choosing the specific options, the user can use any or all of these transcoding methods depending on the limitations of the client or network connection, and can change it when resources change. This makes our proxy very adaptable to serve the varying needs of the user. For example, a user on a laptop may want to only limit the size of video and audio files when s/he is connected via a slow telephone modem, and remove this restriction when connected through the ethernet. A user on a PDA, on the other hand, will want to filter out everything except text and small images.

5. Experimental results

Our proxy server is a modified version of a HTTP server written in Perl. We are using an Apache server to store the preferences and to act as the WWW server capable of content-negotiation. We stored multiple formats of some files and requested them with different preference settings.

An example of transcoding due to content negotiation:

We set the following preferences for one computer (A desktop) Maximum Image file size = 20K Maximum Video file size = 500K

The accept headers added to its request were: Accept: image/x-lgif, image/gif;q=0.6, video/mpeg

We requested for the page http:// bochi. cecs. missouri. edu: 9021/ demo.html

Figure 2 shows the response received.

We set the following preferences for another computer (A laptop) Maximum Image file size = 6K Maximum Video file size = 25K

The accept headers added to its request were: Accept: image/x-sgif, image/gif;q=0.6, video/x-rmpg, video/mpeg;q=0.6

We requested for the same page http:// bochi. cecs. missouri. edu: 9021/ demo.html

Figure 3 shows the response received.

An example of transcoding of images received from the server:

For the same two computers, (same preferences set as above) we requested the page http://www.missouri.edu/~csacm The image file existed only in gif format on the server. The image in the document is small enough for the desktop and hence passed through without any reduction as seen in Figure 4. But the image is large for the laptop (larger than 6K). Therefore, the proxy reduced the resolution of the image as seen in Figure 5.

To see some video files and their representative frames, please visit http://meru.cecs.missouri.edu/~sansanee/mpeg

We kept different settings on two computers and accessed the same web page to see the difference. For the first computer we set a large value for the maximum size of image and video files allowed, and sent accept headers to allow large gif files and mpeg movie files. Therefore, we received large image files and the entire movie file. For the second computer, we limited the size of image and video files, and sent accept headers requesting for small images and representative frames of mpeg files. Hence, we received smaller versions of the images and only the representative frames for the movie file. Our request did not specify the extension of the file name, and the file was available in multiple formats. If the size of the received image files is larger than the maximum size specified, the proxy scales it down either by size or by color as set by the user.

Clearly, the results are best understood by experiencing the proxy based model. We have made the proxy available on the web, it may be accessed at the URL http://nirvana.cecs.missouri.edu:8001. A major drawback of the present implementation is its overhead. Since this is a demonstation prototype, it has been mostly in PERL, and is thus may slow to execute.

6. Discussion

In this paper, we have presented a proxy based system (MOWSER) to support web browsing from mobile platforms. It follows the client-proxy-server model which is the basis of most mobile applications and uses the proxy to provide active transcoding. Proxies are mostly used for forwarding data between the mobile client and the stationary server. The idea of using transcoding at the proxy to support mobility is not new per se. Many proxy based systems [5],[14],[19] have been developed to provide web access to mobile users. However, they typically transcode the image data received from the WWW server before sending it to the mobile client, and are often not configurable. In Mowser, we extend the notion of transcoding to both the upstream and the downstream traffic. More specifically, the upstream request is munged into a HTTP/1.1 request, to make use of the content negotiation feature, and Accept headers are appended, to request for the document in the format most appropriate for the QoS parameters set by the mobile user. On the downstream, in addition to transcoding of images, we also provide the options of removing the active content and transcoding HTML. The mobile user can select any or all of these options. This is to support the changing

requirements of a wide variety of mobile hosts ranging from a powerful notebook with 233MHz CPU, 2GB RAM, which may require only image and video data transcoding, to a PDA with 64KB dynamic RAM, which requires all possible transcoding and filtering of data.

In ongoing work, we are extending the proxy to effectively use all the preferences set by the user to limit or transform the data before serving the MH. Further, our proxy adds a performance overhead due to two reasons. First, it is written in Perl and uses netpbm for the processing of image files. The speed could be increased by writing optimized C code and image conversion routines. Second, messages go all the way up to the application layer in the proxy even if data just needs to be written from one socket to another. Research is going on in the IBM Watson Labs to avoid this by using TCP Splice [15] which allows data to flow through without going to the application layer in the proxy if necessary and such techniques could eventually be integrated into our system. The overhead is justifiable because we are trading proxy side CPU cycles for the more expensive client side CPU cycles and network bandwidth. In experimental situations, it has been observed that extra time taken by the proxy is still less than the time needed to send untransformed data on the wireless network.

References

[1] S. Auephanwiriyakul, A. Joshi and R. Krishnapuram, "Fuzzy Shot Clustering to Support Networked Video Databases," *IEEE FUZZ-IEEE 98/WCCI98* May 1998.

[2] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," *Plenum Press* New York, 1981.

[3] C. Brooks, M. S. Mazer, S. Meeks and J. Miller, "Application-Specific Proxy Servers as HTTP Stream Transducers," *Proc. WWW-4, Boston, http://www.w3.org/pub/Conferences/WWW4/Papers/56* May 1996.

[4] G. Forman and J. Zahorjan, "The Challenges of Mobile Computing," *IEEE Computer, 27:38-47* April 1994.

[5] A. Fox and E. A. Brewer, "Reducing WWW Latency and Bandwidth Requirements by Real-Time Distillations," *Fifth International World Wide Web Conference* May 1996.

[6] IBM Corporation, "Ringing in Wireless Services: Web Access Without Wires," *http://www.ibm.com/Stories/1997/08/wireless4.html*.

[7] Intel Corporation, "Intel Quick Web Technology: White Paper," *http://www.intel.com/quickweb/white.htm*.

[8] A. Joshi, S. Aucphanwiriyakul and R. Krishnapuram, "On Fuzzy Clustering and Content Based Access to Networked Video Databases," *8th IEEE Workshop on Research Issue in Data Engineering* 1998.

[9] A. Joshi, R. Weerasinghe, S. P. McDermott, B. K. Tan, G. Bernhardt and S. Weerawarna, "Mowser: Mobile Platforms and Web Browsers," *Bulletin of the IEEE Technical Committee on Operating Systems and Application Environments* Vol 8, no. 1, 1996.

[10] A. Joshi, S. Weerawarna and E. N. Houstis, "On Disconnected Browsing of Distributed Information," *Proceedings of the seventh International workshop on RIDE* pp 101-107, IEEE Press, 1997.

[11] R. H. Katz, E. A. Brewer, E. Amir, H. Balakrishnan, A. Fox, S. Gribble, T. Hodes, D. Jiang, G. T. Nguyen, V. Padmanabhan, M. Stemm, "The Bay Area Research Wireless Access Network (BARWAN)," *Proceedings Spring COMPCON Conference* 1996.

[12] P. R. Kersten, "Fuzzy Order Statistics and Its Application to Fuzzy Clustering," *Naval Air Warfare Center Weapons division Report* August 1995.

[13] R. Krishnapuram and J. M. Keller, "A Possibility Approach to Clustering," *IEEE Transactions of Fuzzy Systems* vol. 1 No. 2, pp 98-110, May 1993.

[14] M. Liljeberg, H. Helin, M. Kojo, and K. Raatikainen, "Enhanced Services for World-Wide Web in Mobile WAN Environment," *Report C-1996-28* April 1996.

[15] D. Maltz and P. Bhagwat, "MSOCKS: An Architecture for Transport Layer Mobility," *IEEE Infocom 98, San Francisco* pp 1037-1045, April 1998.

[16] B. D. Noble, M. Price, M. Sathyanarayanan, "A Programming Interface for Application-Aware Adaptation in Mobile Computing," *Proceedings of the Second USENIX Symposium on Mobile & Location-Independent Computing* Ann Arbor, MI, April 1995.

[17] B. D. Noble, M. Sathyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, K. R. Walker, "Agile Application-Aware Adaptation for Mobility" *Proceedings of the 16th ACM Symposium on Operating System Principles*

[18] George Vanecek Jr., "Personal Communication," 1995

[19] B. Zenel. "A Proxy Based Filtering Mechanism for The Mobile Environment," *Thesis Proposal, Department of Computer Science, Columbia University*.

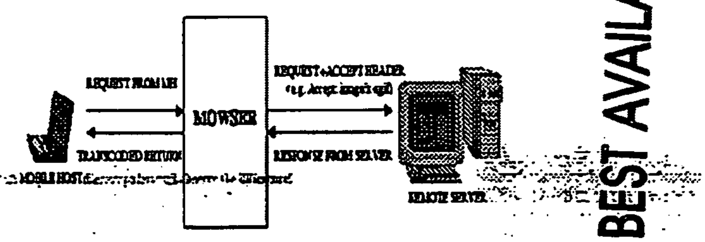

Figure 1. Active Transcoding of HTTP Stream

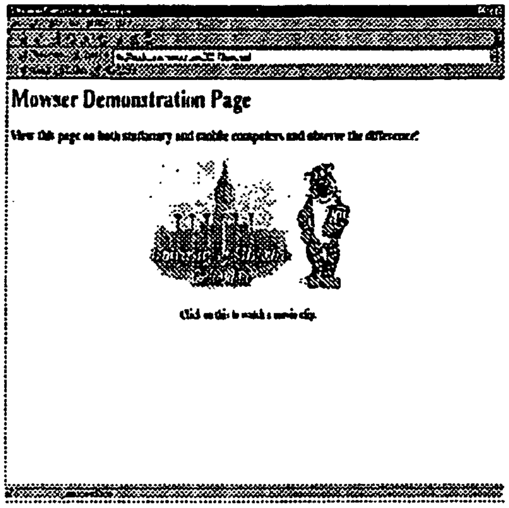

Figure 2. Example of Content Negotiation: Response on a Resource rich Client

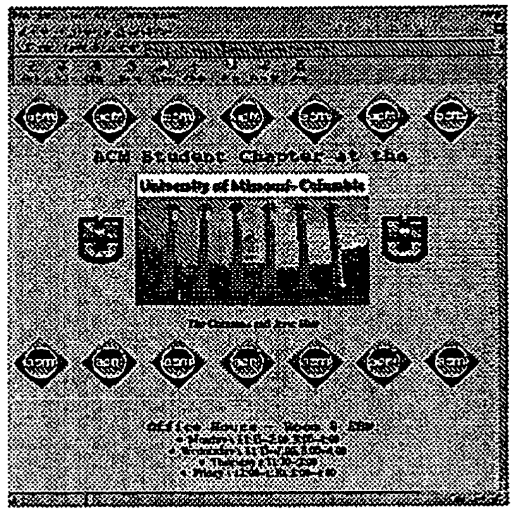

Figure 4. Example of Image Transcoding: Response on a Resource rich Client

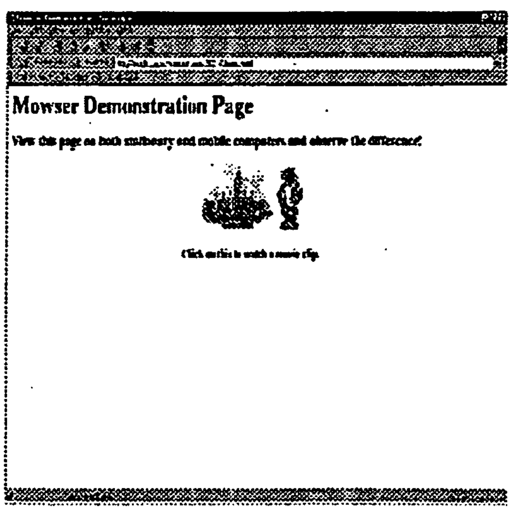

Figure 3. Example of Content Negotiation: Response on a Resource poor Client

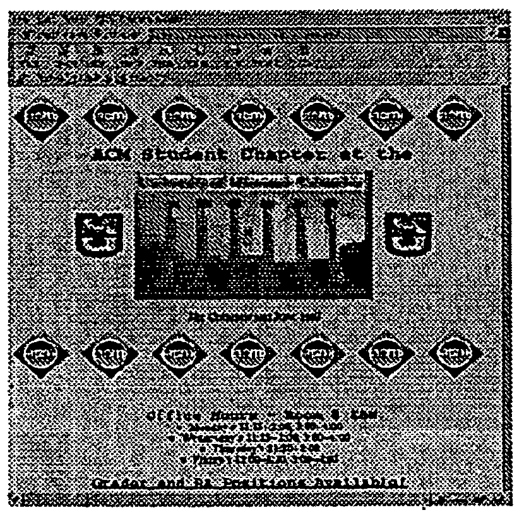

Figure 5. Example of Image Transcoding: Response on a Resource poor Client

TRANSCODING INTERNET CONTENT FOR HETEROGENEOUS CLIENT DEVICES

John R. Smith, Rakesh Mohan and Chung-Sheng Li

IBM T.J. Watson Research Center,
30 Saw Mill River Road, Hawthorne, NY 10532
{jrsmith,rakesh,csli}@watson.ibm.com

ABSTRACT

There is a growing diversity of client devices that have access to the Internet. However, much of the content on the Internet cannot be handled by the devices that have limited communication, processing, storage and display capabilities. In order to improve the utility of a wide range of client devices, we propose a network-based solution for transcoding Internet content. The system uses an InfoPyramid for representing and transcoding video, images, audio and text. The InfoPyramid manipulates the content along the dimensions of fidelity and modality, and aggregates the methods for content analysis, translation, filtering and selection. The InfoPyramid utilizes a policy engine, which incorporates user and publisher preferences, various transcoding policies, device descriptions, and real-time network constraints in order to adapt the Internet content to the client devices.

1. INTRODUCTION

Many new devices, such as personal digital assistants (PDAs), hand-held computers, Internet-ready phones, and WebTVs, are gaining access to the Internet. The capabilities of these devices to receive, process, store and display Internet content varies widely. Given the variety of client devices, it is difficult for Internet content publishers to tailor content to the individual devices.

Internet content publishers do not have many options for customising the content. In some cases, the publishers manually generate secondary, text-only versions of Web pages that the users can select instead of the originals. Other mechanisms within the HTTP protocol allow the client to specify some client attributes, such as the preferred language of the user, or the image, video, and audio formats supported by the client device. Using this information, the content server can automatically select and deliver content that is compatible with the client device and the user.

In the case of client devices with minimal capabilities, such as pagers and cell phones, special markup languages are being developed, such as HDML ([1]). However, mechanisms still need to be developed to automatically convert Internet content into these formats. The emerging XML markup language may improve the capabilities for adapting content since separate style sheets can be developed for client devices that determine how the content is displayed [2].

1.1. Related work

Recently, several systems have been developed for adapting Internet content to client devices. Fox, et al., developed a system for distilling, or compressing images that pass through an Internet proxy [3]. Other commercial systems such as Intel's Quick Web [4] and Spyglass' Prism [5] similarly compress the images that pass through the Internet service provider to speed-up download time.

We have developed a content-based system for transcoding images [6]. This system retrieves and analyses images in the Internet and classifies them into image type and purpose classes in order to select appropriate methods for transcoding them. Since the system is limited to transcoding images, we propose a more powerful solution for Internet content that includes video, images, text and audio. The system transcodes the content along the dimensions of fidelity and modality in order to better adapt it to a large variety of client devices.

1.2. Outline

In this paper, we propose a system for transcoding Internet content. In Section 2, we present an overview of the Internet content transcoding system. In Section 3, we present the InfoPyramid framework for representing and manipulating video, images, audio and text. In Section 4, we describe processes for selecting the versions of the content to adapt the content to the clients according to capabilities and preferences. Finally, in Section 5, we describe the deployment of the transcoding system in the Internet as transcoding proxies.

2. INTERNET CONTENT TRANSCODER

Figure 1 illustrates the proposed Internet content transcoding system. The system retrieves and analyses the Internet content and ingests it into the InfoPyramid format. A policy engine gathers the capabilities of the client, the network conditions and the transcoding preferences of the user and publisher. This information is used to define the transcoding options for the client. The system then selects the output versions of the content and uses a library of content analysis, filtering, translation and manipulation routines to generate the content to be delivered to the client device.

The Internet content transcoding system may be deployed at the server, proxy or client. Deployed at a proxy, the system is able to retrieve the Internet content, analyse

0-7803-4455-3/98/$10.00 © 1998 IEEE

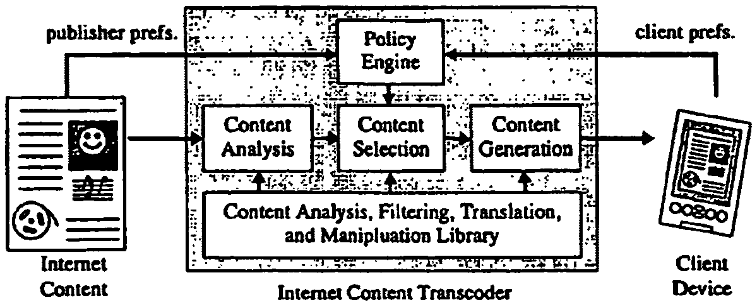

Figure 1: The Internet content transcoding system adapts the content to the capabilities of the client devices.

and transcode it, and deliver the results to the client, on-the-fly. Deployed at the server, the system can be used in the content publication process. The system can prematerialise the alternate versions of the Internet content and store them at the server. In this case, the system merely selects the versions of the content to deliver to the client. In some cases, the transcoding system can be deployed at the client to customise the content display, such as according the user preferences, as long as the client has sufficient capabilities.

3. INFOPYRAMID FRAMEWORK

The InfoPyramid provides a general framework for handling the Internet content. It allows specialized methods to be plugged-in for analyzing, filtering, translating and manipulating the Internet content. As depicted in Figure 2, the InfoPyramid develops a conceptually redundant representation of the Internet content that aggregates multiple versions of the content along the dimensions of modality (video, image, text, and audio) and fidelity (which includes summarised and compressed versions) [7]. The translation and summarisation methods generate the alternate versions of the content as needed.

3.1. Translation and summarisation

The translation methods convert the content between modalities, such as, text to audio, or video to images. On the other hand, the summarisation methods generate versions within the same modality, but with different fidelity. For example, the summarisation methods compress the images, summarise text, and extract and re-animate the key-frames from video. The translation and summarisation methods can be cascaded to change both the modality and fidelity of the content. In this case, a video can be converted to a small, reduced-color image.

For each of the modalities, we describe some of the summarisation and translation methods that could be used to change the fidelity and modality of the content, respectively.

1. Images:
   - Fidelity – Spatial size reduction, color depth reduction, lossy compression [6].
   - Modality – Images to text: related text, embedded text, semantic labels [8].
2. Video:
   - Fidelity – Spatial size, temporal size, playback rate, bit-rate.
   - Modality – Video to images: key-frames; Video to audio: sound track; Video to text: closed captions, embedded text.
3. Text:
   - Fidelity – Key-term extraction, text summarization, document headings extraction.
   - Modality – Text to audio: speech synthesis, Text to text: language translation (i.e., English to French).
4. Audio:
   - Fidelity – Bit-rate reduction, stereo to mono.
   - Modality – Audio to text: speech recognition.

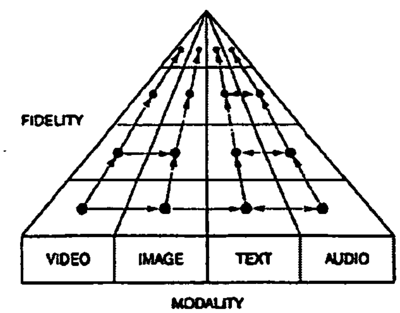

Figure 2: The InfoPyramid aggregates multiple representations of the content along the dimensions of fidelity and modality and unifies the methods for manipulating the content.

In order to evaluate content alternatives, the system could assign content value scores to each of the content

alternatives. Using the content value scores, the system is able to optimise the selection of the content according to the device capabilities, preferences and network conditions.

4. TRANSCODING SYSTEM

We propose that the transcoding system utilises a policy engine to evaluate the alternatives for adapting the content to the client devices.

4.1. Policy Engine

The policy engine would gather the capabilities of the client and the transcoding preferences of the user and publisher, and sense the network conditions to define the transcoding options for the client. In order adapt the Internet content to these devices, the transcoding proxy generates and selects versions of the content according to the policies, network and device constraints, and preferences.

4.1.1. Content value scores

The InfoPyramid system provides the mechanism for assigning content value scores to the alternate versions of the content. In some cases, the content value scores are derived automatically by measuring the loss in fidelity that results from translating or summarising the content. For example, the content value scores can be linked to the distortion introduced from compressing the images or audio. Otherwise, the content value scores can be tied directly to the methods that manipulate the content.

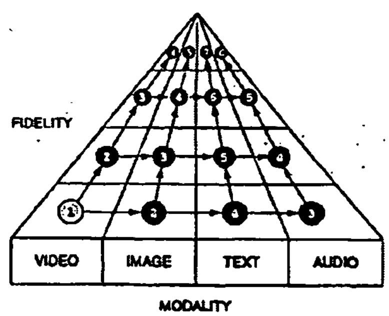

Figure 3: Example Internet content value scores assigned for video.

Figure 3 illustrates examples of content value scores that can be assigned for transcoding video. In this example, the original video has the highest content value. Each manipulation of the video along the dimension of fidelity or modality alters the content value. For example, converting the video to a sequence of images results in a small reduction in content value. Converting the video to a highly compressed audio track produces a higher reduction in the content value.

4.1.2. User and publisher preferences

The content value scores comprise only part of the information that can be used in the content selection process. Both the publisher and user may have preferences for how the content is transcoded. Figure 4 illustrates some example preferences for transcoding images. In this example, the preferred versions for the image content are the low-fidelity versions of the image, and the translations to text.

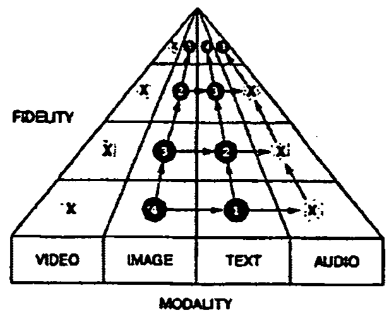

Figure 4: Internet content format preferences may be established by the publisher or user. This example illustrates the preferred content alternatives for images. Device constraints may further score or eliminate some content alternatives.

4.1.3. Device constraints

Various constraints of the devices affect the selection of the content. For example, many devices cannot handle video. In this case, the corresponding content alternatives can be eliminated (see Figure 4). Overall, the display, storage and processing capabilities of the client devices eliminate the selection of individual versions of the content. These also place constraints on the set of selections for a Web document. For example, if the device has a local storage of only 16 KB, then this places a hard limit on the total size of the versions of the content selected.

4.1.4. Network constraints

Similarly, the network would place constraints on the content selection process. In general, the network constraints introduce a trade-off between content data size and load time. For example, if the user specifies a maximum load time for a page, then to accommodate this load time, the transcoder system must sense the end-to-end bandwidth to derive the target data size for the content. Then, the system can select the content of maximum content value that is within the target data size.

4.2. Content Selection

We investigate more closely how the system could optimise the selection of the content alternatives. Consider the Web document with two objects, A and B, depicted in Figure 5. Let $A_{ij} = T_{ij}(A)$ and $B_{kl} = T_{kl}(B)$ transcode A and B,

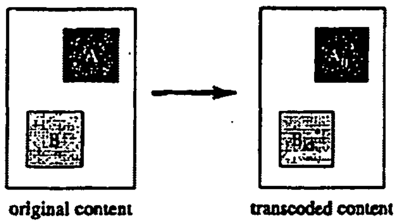

Figure 5: Example of content selection for a Web document consisting of two objects, A and B.

respectively, to target modalities i and k, and fidelities j and l. Let $V(A_{ij})$ and $V(B_{kl})$ be the content value scores for A_{ij} and B_{kl}, and let $S(A_{ij})$ and $S(B_{kl})$ be the data sizes of A_{ij} and B_{kl}. Finally, let S_T be the maximum data size for the content, which may have been derived from the user's specified maximum load-time and the network conditions.

4.2.1. Maximum content value

The content selection process selects A^*_{ij} and B^*_{kl} with maximum content value for a target data size as follows:

$$\begin{aligned} V(A^*_{ij}) + V(B^*_{kl}) &= \max_{i,j,k,l}(V(A_{ij}) + V(B_{kl})), \text{ and} \\ S(A^*_{ij}) + S(B^*_{kl}) &\leq S_T. \end{aligned} \quad (1)$$

4.2.2. Minimum load time

Given a minimum acceptable content value V_T, the content alternatives A^*_{ij} and B^*_{kl} of minimum data size are given by:

$$\begin{aligned} S(A^*_{ij}) + S(B^*_{kl}) &= \min_{i,j,k,l}(S(A_{ij}) + S(B_{kl})), \text{ and} \\ V(A^*_{ij}) + V(B^*_{kl}) &\geq V_T. \end{aligned} \quad (2)$$

4.2.3. Device constraints and preferences

By extending this optimisation process, the content selection system could incorporate the user ($U(A^k_{ij})$) and publisher ($P(A^k_{ij})$) preferences, and client device constraints ($D(A^k_{ij})$), to best adapt the content. In this case, the total score of each item k in the Web document is given by

$$Z(A^k_{ij}) = \omega_v V^k_{ij} + \omega_u U^k_{ij} + \omega_p P^k_{ij} + \omega_d D^k_{ij},$$

where ω_v, ω_u, ω_p, and ω_d are weighting factors assigned by the user or system.

5. REAL-TIME, NETWORK-BASED TRANSCODING

The transcoding system can be implemented in the network in the form of a transcoding proxy (TP) to perform transcoding on-the-fly, see Figure 6. The transcoding proxy system is designed to have a high bandwidth connection to the content provider. In most cases, the proxy has a low bandwidth connection to the client. As a result, reducing the amount data through compression and transcoding at the proxy results in faster delivery, even when accounting for the added time for content analysis, selection and transcoding [6].

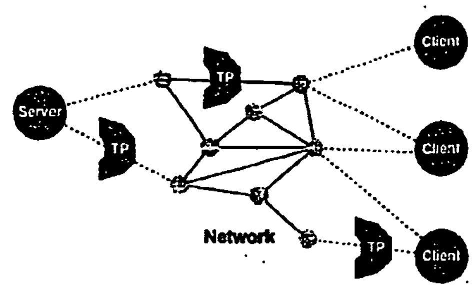

Figure 6: Network-based transcoding of Internet content using transcoding proxies (TP).

6. SUMMARY

We proposed a system for the network-based transcoding of Internet content in order to improve the accessibility of a wide range of client devices to the content on the Internet. The transcoding system retrieves, analyses, and ingests the Internet content into an InfoPyramid representation. The InfoPyramid provides a conceptual framework for manipulating the content along the dimensions of modality and fidelity. The transcoding system selects the content from the InfoPyramid by assessing the various content alternatives to adapt the Internet content to the client devices. In this way, a wide range of client devices gain access to the large amounts of content on the Internet that is otherwise beyond their capabilities.

7. REFERENCES

[1] Unwired Planet. Handheld device markup language specification. Technical Report Version 2.0, Unwired Planet, Inc., April 1997.

[2] C. M. Sperberg-McQueen T. Bray, J. Paoli. Extensible markup language (XML), W3C recommendation. Technical Report Version 1.0, W3C, February 1998.

[3] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir. Adapting to network and client variability via on-demand dynamic distillation. In ASPLOS-VII, Cambridge, MA, October 1996.

[4] Intel Quick Web. http://www.intel.com/quickweb.

[5] Spyglass-Prism. http://www.spyglass/products/prism.

[6] J. R. Smith, R. Mohan, and C.-S. Li. Content-based transcoding of images in the Internet. In IEEE Proc. Int. Conf. Image Processing (ICIP), Chicago, Il, October 1998. Submitted.

[7] C.-S. Li, R. Mohan, and J. R. Smith. Multimedia content description in the InfoPyramid. In IEEE Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP), Seattle, WA, May 1998. Special session on Signal Processing in Modern Multimedia Standards.

[8] J. R. Smith and S.-F. Chang. Visually searching the Web for content. IEEE Multimedia Mag., 4(3):12 – 20, July–September 1997.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Organization TC2100 Bldg./Room PK2

U. S. DEPARTMENT OF COMMERCE
COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
IF UNDELIVERABLE RETURN IN TEN DAYS

OFFICIAL BUSINESS

RETURNED TO SENDER

Del. As Addressed
Unable to Fwd.
- [ ] Attempted not known
- [ ] No such number/street
- [ ] No Receptacle [ ] Refused
- [ ] Vacant [ ] Temp Away
- [ ] Other

AN EQUAL OPPORTUNITY EMPLOYER

UNITED STATES POSTAGE
U.S. OFFICIAL MAIL
PENALTY FOR PRIVATE USE $300
PITNEY BOWES
02 1A
0004205065
$ 01.75⁰
AUG16 2005
MAILED FROM ZIP CODE 22314

RECEIVED
AUG 2 3 2005
USPTO MAIL CENTER